VANGUARD Microarray Study - remove main carinas

Li Shen, Jing Wang, and Kevin R. Coombes
Dept. of Bioinformatics and Computational Biology
The University of Texas M.D. Anderson Cancer Center (MDACC)

August 15, 2012

Contents

1 Executive Summary

The original analysis of this report was done by Dr. Coombes. The report is sorted under: //Mdadqsfs02/bioinfo2/Lung-HN/KRC-Analyses
(01-vanguard-krc.pdf)
In this report, we rerun the code to find genes differentially expressed in a site and time-dependent manner after **removing the main carinas**.

1.1 Introduction

This dataset was acquired using Affymetrix Human Gene ST 1.0 Exon Arrays. The samples came from 19 non-small-cell lung cancer (NSCLC) patients and were collected via bronchial brushings at different sites and different times.

1.1.1 Aims/Objectives

The primary goal is to discover genes and pathways that change over time or that have differential expression based on the proximty of the collection site and the primary tumor.

1.2 Methods

1.2.1 Description of the Data

The processed microarray data consists of measurements of the expression of 32,252 probes in 391 samples from 19 patients. The data were colected in two large batches, which in this case represents different laboratories. The first batch of 70 samples were run in Li Mao's laboratory, while the second batch of 321 samples was run in Ignacio Wistuba's laboratory. Fortunately, the batches are not completely confounded with contrasts of interest, but the first batch is "enriched" for samples collected at the baseline time point and for samples from the main carina.

1.2.2 Statistical Methods

Microarray data were processed using the RMA algorithm as implemented in the aroma.affymetrix package of R. We use a mixed-effects model on the processed data to understand the expression patterns of each gene. the fixed effects represent

- Batch: array data were collected (over time) in two batches from two different laboratories. This effect is a nuisance and not one we really want to use to make inferences.
- DetMap: This is a detailed map of the site where each (bronchial brushing) sample was obtained. There are four levels of this variable, basically corresponding to its distance from the site of the primary tumor. In order, the levels are ADJ (adjacent to the tumor), NON-ADJ (same half of the lung but not adjacent to the tumor), MC (main carina), and CONTRA (from the contralateral lung).
- Time.point: Samples were collected at four time points (0, 12, 24, or 36 months).

- We also allow an interaction term between DetMap and Time.point.

Random effects primarily account for the fact that we have multiple samples from the same patient (Case). We try to fit a model that includes a different starting point and slope over time for each case. For some probes, we are unable to fit a model that includes different slopes for each patient; in this case, we fall back to a single (base-level) random effect foe each case.

1.3 Results

- Batch is a dominant effect on gene expression (**Figure 1**).
- After adjusting for batch (as part of the mixed-effects model), both site and time are significant in some genes, with site more important than time (**Figure 1**).
- There is very weak (and perhaps no real) evidence that the interaction term is ever significant (**Figure 1**, **Figure 3**).

1.4 Conclusions

2 Loading the Data

2.1 R Packages

We start by loading the packages that will be needed for our analysis.

```
> require(nlme)
> library(lattice)
> library(RColorBrewer)
> library(ClassComparison)
> library(ClassDiscovery)
```

We modify some of the default display options for the trellis/lattice plots.

```
> x <- trellis.par.get("plot.symbol")
> x$pch <- 16
> x$col <- "#00aa60"
> trellis.par.set("plot.symbol", x)
> rm(x)
```

2.2 Affymetrix data

The processed Affymetrix data can be found in the following location:

```
> basedir <- ifelse(.Platform$OS == "windows", "//mdadqsfs02", "/data")
> datadir <- file.path(basedir, "bioinfo2",
+                      "Lung-HN", "Wistuba-VANGUARD", "Analysis")
```

Here we load the data.

```
> load(file.path(datadir, "gExpr-RMA-Aroma.RData"))
```

In order to understand what is contained in the dataset, we explore the objects that we just loaded.

```
> ls()
```

```
 [1] "adjData"       "annot"         "basedir"       "batch.col"     "batch.colors"
 [6] "bbat"          "binter"        "bp"            "bsite"         "btime"
[11] "case.col"      "case.colors"   "colInd"        "colorfacs"     "colorfacs.sp"
[16] "colorfacs.sp2" "colt"          "cr"            "cutter"        "datadir"
[21] "ddc"           "ddr"           "ev.col"        "ev.colors"     "expected"
[26] "fixcoef"       "gc"            "gene.label"    "gExpr"         "ggc"
[31] "group"         "hist.col"      "hist.colors"   "keysize"       "labR"
[36] "labS"          "lhei"          "lmat"          "lp"            "lwid"
[41] "m"             "margins"       "mc"            "modlist"       "my.col"
[46] "my.fig"        "my.i"          "my.pcc"        "n"             "n1"
[51] "n2"            "normData"      "observed"      "onesig"        "opar"
[56] "pvals"         "results"       "rmMC.sclass"   "rowInd"        "sc"
[61] "sclass"        "scut"          "scut.colors"   "select.p"      "si"
[66] "simplify"      "si.sp"         "site.col"      "site.colors"   "site.specific"
[71] "sp.case.col"   "sp.site.col"   "ss"            "ssc"           "ssel"
[76] "ssite"         "stime"         "tab"           "tab.site"      "tab.time"
[81] "tab.time2"     "tclass"        "tcut"          "tcut.colors"   "temp"
[86] "temp.s"        "tf"            "time.col"      "time.colors"   "time.lapse"
[91] "tsel"          "tt"            "ulim"
```

```
> class(gExpr)
```

```
[1] "data.frame"
```

```
> dim(gExpr)
```

```
[1] 33252   396
```

```
> class(normData)
```

```
[1] "data.frame"
```

```
> dim(normData)
```

```
[1] 33252   391
```

```
> normData <- as.matrix(normData)
> class(si)
```

```
[1] "data.frame"

> dim(si)

[1] 391  61

> all(colnames(normData)==rownames(si))

[1] TRUE

> table(si$DetMap)

            ADJ  CONTRA      MC NON-ADJ
      1      62     161      60     107
```

Here we remove the main carinas.

```
> mc <- which(si$DetMap %in% "MC")
> si <- si[-mc,]
> normData <- normData[,-mc]
> dim(si)

[1] 331  61

> dim(normData)

[1] 33252   331

> all(colnames(normData)==rownames(si))

[1] TRUE

> table(si$DetMap)

            ADJ  CONTRA NON-ADJ
      1      62     161     107

>
```

2.3 Gene/Probe Annotations

Here we load the annotations for the probes on the ST 1.0 array.

```
> annot <- read.csv(file.path(datadir, "RNW",
+                             "Human Gene 1.0 ST annotations for Li Shen.csv"),
+                   header=TRUE, as.is=TRUE, na.strings=c("NA","Un", ""),
+                   row.names=1)
> all(rownames(normData) %in% rownames(annot))
```

```
[1] TRUE
```

```
> annot <- annot[rownames(normData),]
> annot$Symbol <- factor(annot$Symbol)
> annot$UGCluster <- factor(annot$UGCluster)
> annot$Chromosome <- factor(annot$Chromosome,
+                            levels=c(1:22, "X", "Y"))
> annot$Cytoband <- factor(annot$Cytoband)
> summary(annot)
```

```
     Name            Accession            UGCluster           Symbol
 Length:33252       Length:33252       Hs.559040:   26   LOC349196:   26
 Class :character   Class :character   Hs.199343:   13   DUX4     :   12
 Mode  :character   Mode  :character   Hs.196086:   12   FAM90A1  :   12
                                       Hs.460179:   12   MGC72080 :   12
                                       Hs.553518:   12   SMG1     :   12
                                       (Other)  :21386   (Other)  :21697
                                       NA's     :11791   NA's     :11481
   EntrezID           Chromosome       Cytoband          GO
 Min.   :        1   1      : 2241   6p21.3 :  356   Length:33252
 1st Qu.:     7166   19     : 1442   19p13.3:  210   Class :character
 Median :    51585   2      : 1385   16p13.3:  203   Mode  :character
 Mean   :  1588044   11     : 1374   19p13.2:  170
 3rd Qu.:   124778   6      : 1355   Xq28   :  133
 Max.   :100499221   (Other):13971   (Other):20629
 NA's   :    11481   NA's   :11484   NA's   :11551
```

2.4 Clinical Data

Now we clean up the clinical data.

```
> si$Batch <- factor(si$Batch)
> si$Gender <- factor(si$Gender)
> si$Diagnosis..Histology. <- factor(si$Diagnosis..Histology.)
> colnames(si)[19] <- "Histology"
> si$Off.study_Reason <- factor(si$Off.study_Reason)
> si$Differentiation <- factor(si$Differentiation)
> si$Leison.Site <- factor(si$Leison.Site)
> si$Anatomical_site <- factor(si$Anatomical_site)
> si$site.of.collection <- factor(si$site.of.collection)
> si$Contralateral <- factor(si$Contralateral)
> dmlev <- c("ADJ", "NON-ADJ", "CONTRA")
> si$DetMap <- factor(si$DetMap, levels=dmlev)
> mlev <- c("ADJ", "NON-ADJ")
```

```
> si$Map <- factor(si$Map, levels=mlev)
> #si$Time.point <- as.numeric(si$Time.point)
> si$Code.4.time.point <- factor(si$Code.4.time.point)
> si$Code.4.Site.of.collection <- factor(si$Code.4.Site.of.collection)
> si$pT <- factor(si$pT)
> si$pN <- factor(si$pN)
> si$Final.Pat.Stage <- factor(si$Final.Pat.Stage)
> si$EGFR.status <- factor(si$EGFR.status)
> si$KRAS.status <- factor(si$KRAS.status)
> si$V_Case.ID.Inclusion_number. <- factor(paste("P",
+                                                si$V_Case.ID.Inclusion_number.,
+                                                sep=''))
> colnames(si)[8] <- "Case"
> colnames(si)[13] <- "MRN"
> simplify <- c(2, 8, 27, 31, 34, 12, 14:20, 22, 24, 43:47, 13)
> rm(dmlev, mlev)
> summary(si[, simplify])
```

```
 Batch         Case         DetMap         Map        Time.point
 I : 37   P1     : 20   ADJ    : 62   ADJ    : 62   Min.   : 0.00
 II:294   P20    : 20   NON-ADJ:107   NON-ADJ:268   1st Qu.: 0.00
          P3     : 20   CONTRA :161   NA's   :  1   Median :12.00
          P30    : 20   NA's   :  1                 Mean   :15.73
          P31    : 20                               3rd Qu.:24.00
          P40    : 20                               Max.   :36.00
          (Other):211
          Off.study_Reason Gender  DOB..DOBirth.       DOSurgery
                  : 14     F:113   Length:331         Length:331
 N                :282     M:218   Class :character   Class :character
 Y_died           : 20             Mode  :character   Mode  :character
 Y_recurrence lung: 15
```

```
 DOInclusion          Surgery                   Histology   Differentiation
 Length:331         Length:331         Adenocarcinoma:261   MOD     :151
 Class :character   Class :character   Squamous      : 70   MOD-POOR: 20
 Mode  :character   Mode  :character                        POOR    : 31
                                                           W       : 35
                                                           WELL    : 20
                                                           NA's    : 74
```

```
 Anatomical_site Contralateral    pT      pN      Final.Pat.Stage      EGFR.status
```

```
 LLL: 48               :  1    1A  :170   0:311   I  : 20         MUT del E19: 15
 LUL: 86         CONTRA:161    1B  : 35   1: 20   IA :185         WT         : 59
 RLL: 68         IPSI  :169    2A  : 86           IB : 86         NA's       :257
 RML: 15                       2B  : 20           IIA: 40
 RUL:114                       NA's: 20


     KRAS.status
 MUT cod 12: 20
 WT        : 54
 NA's      :257
```

```
> tmp <- si[match(levels(as.factor(si$MRN)), si$MRN), c(8,13)]
> tmp2 <- read.csv(file.path(datadir, "RNW",
+                            "Copy of VANGUARD_06012011_BRONCHIAL BRUSHES.csv"),
+                  header=TRUE, as.is=TRUE, na.strings=c("NA","Un", ""),
+                  row.names="MDAH")
> ci <- data.frame(tmp, Event=tmp2[match(tmp$MRN, rownames(tmp2)),"Event"])
> ci <- ci[order(ci$Case),]
> rownames(ci) <- ci$Case
> Event.col <- rep("RecurrenceII", nrow(ci))
> Event.col[which(ci$Event=="YES")] <- "RecurrenceI"
> Event.col[which(ci$Event=="NO")]  <- "No"
> ci <- cbind(ci, Event.col)
> ci <- ci[, c(1,4)]
> foo <- merge(si, ci, by="Case")
> for (i in 1:nrow(foo)) {
+   w <- which(si$Experiment.Names == foo[i, 'Experiment.Names'])
+   if (length(w) != 1)
+     stop("no unique match")
+   rownames(foo)[i] <- rownames(si)[w]
+ }
> foo <- foo[rownames(si),]
> foo <- foo[,c(colnames(si), "Event.col")]
> all(si[, 1:61]==foo[,1:61])

[1] NA

> si <- foo
> rm(foo, ci, Event.col, tmp, tmp2, i, w)
> write.table(si, file="SampleInfo.csv", sep=",")
```

These colors will be used in some of the later plots.

```
> ev.col <- c(No="white", RecurrenceI="black", RecurrenceII="gray")
> ev.colors <- ev.col[as.character(si$Event.col)]
> hist.col <- c(Adenocarcinoma='orange',
+               Squamous='purple')
> hist.colors <- hist.col[as.character(si$Histology)]
> batch.col <- c(I='cyan',
+                II='magenta')
> batch.colors <- batch.col[as.character(si$Batch)]
> site.col <- brewer.pal(5, "Reds")[2:5]
> names(site.col) <- levels(si$DetMap)
> site.col <- c(ADJ="red", "NON-ADJ"="gold", CONTRA="blue")
> site.colors <- site.col[as.numeric(si$DetMap)]
> time.col <- brewer.pal(5, "Blues")[2:5]
> names(time.col) <- seq(0, 36, 12)
> time.colors <- time.col[1+round(si$Time.point/12)]
> case.col <- c(brewer.pal(3, "Reds"),
+               brewer.pal(3, "Blues"),
+               brewer.pal(3, "Greens"),
+               brewer.pal(3, "Purples"),
+               brewer.pal(3, "Greys")[2:3],
+               brewer.pal(12, "Paired")[11],
+               brewer.pal(9, "Set1")[6:7],
+               brewer.pal(8, "Dark2")[4],
+               "#88e1e1")
> names(case.col) <- levels(si$Case)
> case.colors <- case.col[as.numeric(si$Case)]
> #barplot(rep(1, 19), col=case.col)
> colorfacs <- list(
+                   Case=list(
+                     fac=si$Case,
+                     col=case.col),
+                   Site=list(
+                     fac=si$DetMap,
+                     col=site.col),
+                   Time=list(
+                     fac=factor(si$Time.point),
+                     col=time.col),
+                   Batch=list(
+                     fac=si$Batch,
+                     col=batch.col),
+                   Histology=list(
+                     fac=si$Histology,
```

```
+                     col=hist.col),
+                   Event=list(
+                     fac=si$Event.col,
+                     col=ev.col)
+                   )
> cr <- colorRampPalette(c("white", brewer.pal(9, "Oranges")))
> tf <- function(x) x^0.15
```

3 Statistical Modeling

We use a mixed-effects model to understand the expression patterns of each gene. Fixed effects represent

- Batch: array data were collected (over time) in two batches from two different laboratories. This effect is a nuisance and not one we really want to use to make inferences.
- DetMap: This is a detailed map of the site where each (bronchial bruishing) sample was obtained. There are four levels of this variable, basically corresponding to its distance from the site of the primary tumor. In order, the levels are ADJ (adjacent to the tumor), NON-ADJ (same half of the lung but not adjacent to the tumor), MC (main carina), and CONTRA (from the contralateral lung).
- Time.point: Sample were collected at four time points (0, 12, 24, or 36 months).
- We also allow an interaction term between DetMap and Time.point.

Random effects primarily account for the fact that we have multiple samples from the same patient (Case). We try to fit a model that includes a different starting point and slope over time for each case. For some genes, we are unable to fit a model that includes different slopes for each patient; in this case, we fall back to a single (base-level) random effect for each case.

```
> gene.label <- function(gene) {
+   ifelse(is.na(annot[gene, "Symbol"]),
+          rownames(annot)[gene],
+          as.character(annot[gene, "Symbol"]))
+ }
> f <- "modlist.rda"
> if (file.exists(f)) {
+   load(f)
+ } else {
+   modlist <- lapply(1:nrow(normData), function(x) 1)
+   for (gene in 1:nrow(normData)) {
+     gl <- gene.label(gene)
+     cat(gl, "\n", file=stderr())
+     pinfo <- 1:nrow(si)
+     pclin <- si[pinfo, simplify]
+     x <- normData[gene, pinfo]
+     tempd <- data.frame(si[, c(2,8, 27, 34, 19)], Y=x)
+     foo <- na.omit(tempd)
+     foo$Time.point <- foo$Time.point/12
+     foo <- foo[order(foo$Time.point, foo$Case),]
+     gd <- groupedData(Y ~Time.point|Case,data=foo, outer=~Histology)
+     mod6 <- try(lme(Y ~ Batch + DetMap*Time.point, data=gd,
```

```
+                     random = ~ Time.point|Case,
+                     method="ML"))
+     if (inherits(mod6, "try-error")) {
+       mod6 <- (lme(Y ~ Batch + DetMap*Time.point, data=gd,
+                    random = ~ 1|Case,
+                    method="ML"))
+     }
+     modlist[[gene]] <- mod6
+   }
+   rm(gene, gl, pinfo, pclin, x, tempd, foo, gd, mod6)
+   save(modlist, file=f)
+ }
> rm(f)
```

3.1 Relative Importance of Fixed Effects

In the next block of code, we extract the p-values from the statistical models. To illustrate what we expect to get, we first show an example.

```
> x <- modlist[[1]]
> anova(x)
```

This example shows that we get separate p-values for each of the fixed effects included in the model, stored as the fourth column of the ANOVA table. So, we extract that column for each gene and save it.

```
> f <- "pvals.rda"
> if (file.exists(f)) {
+   load(f)
+ } else {
+   lap <- lapply(modlist, function(x) {
+     a <- anova(x)
+     a[,4]
+   })
+   pvals <- matrix(unlist(lap), ncol=5, byrow=TRUE)
+   a <- anova(modlist[[1]])
+   colnames(pvals) <- rownames(a)
+   rownames(pvals) <- rownames(normData)
+   pvals <- as.data.frame(pvals)
+   rm(lap, a, x)
+   save(pvals, file=f)
+ }
> rm(f)
```

We fit beta-uniform-mixture (BUM) models to the p-values for each of the four fundamental terms in the statistical model. We also plot histograms for the distributions of these p-values (**Figure 1**). It is clear that batch is an extremely large effect, being present in almost every gene. However, after adjusting for batch, both the site and the time produce clear signs of changing (for some genes) across the samples in a consistent manner, with site being slightly more important than time.

```
> bsite <- Bum(pvals$DetMap)
> countSignificant(bsite, alpha=0.01)

[1] 136

> btime <- Bum(pvals$Time.point)
> countSignificant(btime, alpha=0.01)

[1] 502

> bbat <- Bum(pvals$Batch)
> countSignificant(bbat, alpha=0.01)

[1] 22308

> binter <- Bum(pvals$"DetMap:Time.point")
> countSignificant(binter, alpha=0.01)

[1] 0
```

3.2 Interaction Between Time and Site

Next, we would like to better understand the interaction term in the model. The histogram for the p-values associated with the interaction is a slightly odd shape, in that the standard BUM model clearly does not fit the distribution (**Figure 1**). This observation should not be terribly surprising, since the model with an interation term only makes sense if the main effects (site and time) are themselves significant.

We start by asking whether the significance of site and time is correlated (tends to happen for the same genes) or independent. A smooth scatter plot of the logistically transsformed p-values strongly suggests that they are independent (**Figure 2**). Directly counting the overlap at a 5% significance level agrees with this assessment.

```
> ss <- countSignificant(bsite, alpha=0.05)
> ss

[1] 701
```

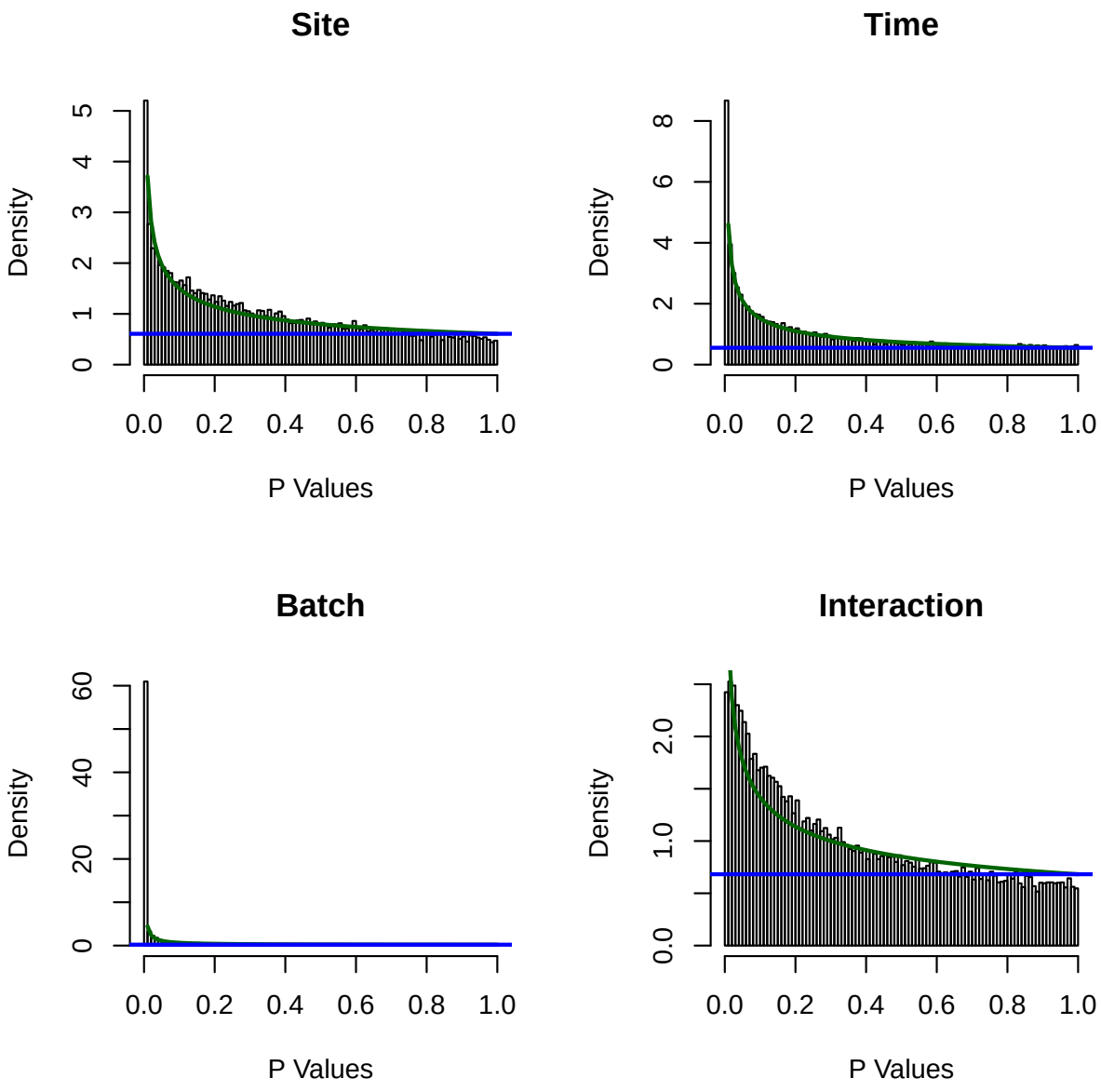

Figure 1: Histograms of p-values for the fixed effects.

```
> tt <- countSignificant(btime, alpha=0.05)
> tt

[1] 2381

> observed <- sum(selectSignificant(bsite, alpha=0.05) &
+                 selectSignificant(btime, alpha=0.05))
> expected <- ss*tt/nrow(normData)
> round(c(OBS=observed, EXP=expected))

OBS EXP
 40  50
```

Now we restrict to the set of genes where there is some very weak evidence that both time and site have significant effects. There are about 3000 probes for which both time and site have $p < 0.20$.

```
> cutter <- 0.20
> onesig <- selectSignificant(bsite, alpha=cutter) &
+     selectSignificant(btime, alpha=cutter)
> sum(onesig)

[1] 1685
```

We can fit a BUM model to the interaction p-values associated with this subset of probes. We still get a fairly small number of genes, even with a 30% FDR.

```
> bp <- Bum(pvals$"DetMap:Time.point"[onesig])
> countSignificant(bp, alpha=0.30)

[1] 150
```

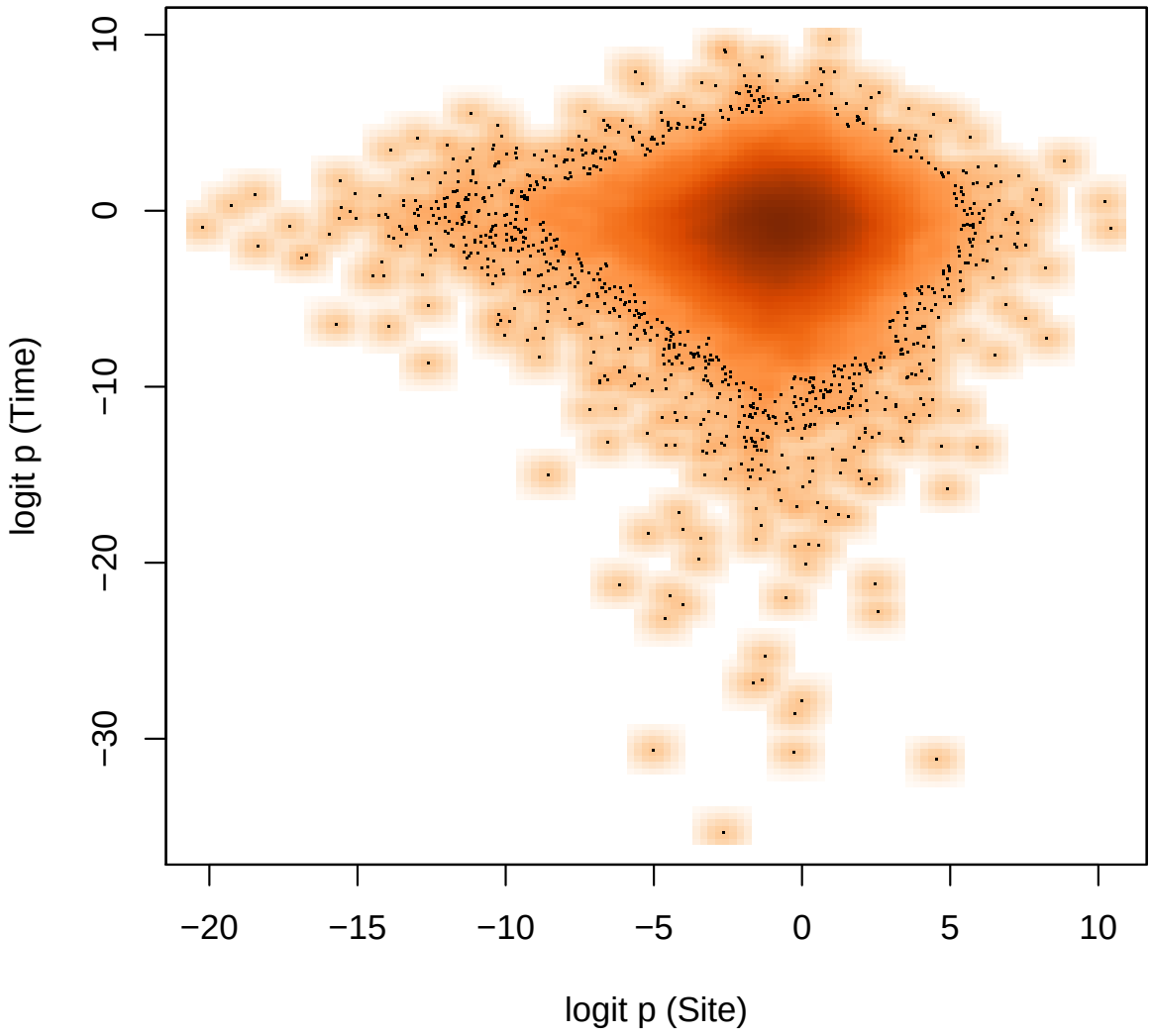

Figure 2: Smoothed scatter plot logistically transformed p-values from site and time.

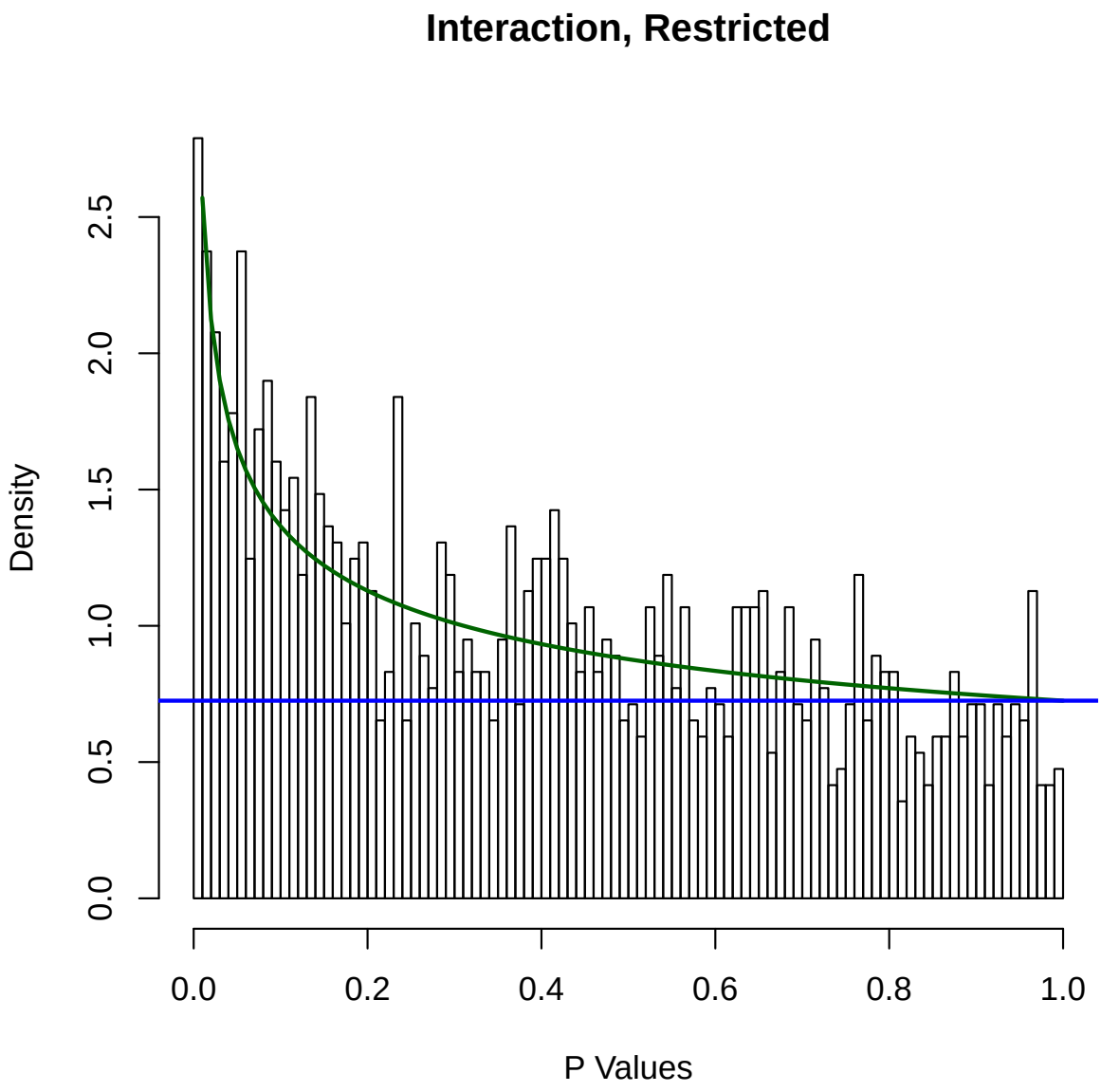

Figure 3: Histogram of p-values for the interaction between time and site, restricted to genes where both main effects show a trend toward significance.

4 Batch Correction

In order to generate additional plots, we need to adjust for the batch effects that are imposed on almost all genes. The information that we need to make this adjustment is already contained in the (fixed-effects) coefficients in the statistical models that we computed for each gene. For example,

```
> x <- modlist[[1]]
> fixef(x)
```

The next block of code extracts all of the fixed-effects coefficients from the statistical models.

```
> f <- "fixcoef.rda"
> if (file.exists(f)) {
+   load(f)
+ } else {
+   fip <- lapply(modlist, fixef)
+   fixcoef <- matrix(unlist(fip), ncol=7, byrow=TRUE)
+   colnames(fixcoef) <- names(fixef(x))
+   rownames(fixcoef) <- rownames(normData)
+   fixcoef <- as.data.frame(fixcoef)
+   rm(fip, x)
+   save(fixcoef, file=f)
+ }
> rm(f)
```

Now we use the batch coefficients to adjust the data.

```
> adjData <- normData
> temp <- sweep(adjData[, si$Batch=="I"], 1, fixcoef$BatchII, "+")
> adjData[, si$Batch=="I"] <- temp
```

4.1 Genes That Are Different By Site

We start by selecting the genes that are signifcantly different between sites, based on a 1% false discovery rate (FDR).

```
> ssel <- selectSignificant(bsite, alpha=0.01)
> site.specific <- adjData[ssel,]
> ggc <- hclust(distanceMatrix(t(site.specific), "pearson"), "ward")
> scut <- cutree(ggc, k=2)
> scut.colors <- brewer.pal(8, "Dark2")[scut]
> sclass <- as.numeric(ssel)
> sclass[ssel] <- scut
> table(sclass)
```

```
sclass
    0     1     2
33116    23   113
```

```
> results <- data.frame(site.specific,
+                       DetMap=pvals[ssel, "DetMap"],
+                       SiteClass=sclass[ssel],
+                       annot[ssel,])
> results <- results[order(results$DetMap, decreasing=F),]
> write.csv(results, file=file.path(datadir, "site.specific.csv"))
> rmMC.sclass <- sclass
> save(rmMC.sclass, file="01-1-rmMCsclass-Site.R")
>
```

Now we cluster the samples using these genes (**Figure 4**).

```
> ssc <- hclust(distanceMatrix(site.specific, "pearson"), "ward")
```

There are more "adjacent" samples in the left branch and more "main carina" and "contralateral" samples in the right branch of the dendorgram; this difference is statistically significant.

```
> table(cutree(ssc, k=2))

  1   2
103 228
```

```
> tab.site <- table(cutree(ssc, k=2), si$DetMap)
> tab.site

    ADJ NON-ADJ CONTRA
  1  32      33     37
  2  30      74    124
```

```
> round(tab.site[1,]/apply(tab.site, 2, sum)*100, 1)

    ADJ NON-ADJ  CONTRA
   51.6    30.8    23.0
```

```
> fisher.test(tab.site)

        Fisher's Exact Test for Count Data

data:  tab.site
p-value = 0.0002656
alternative hypothesis: two.sided
```

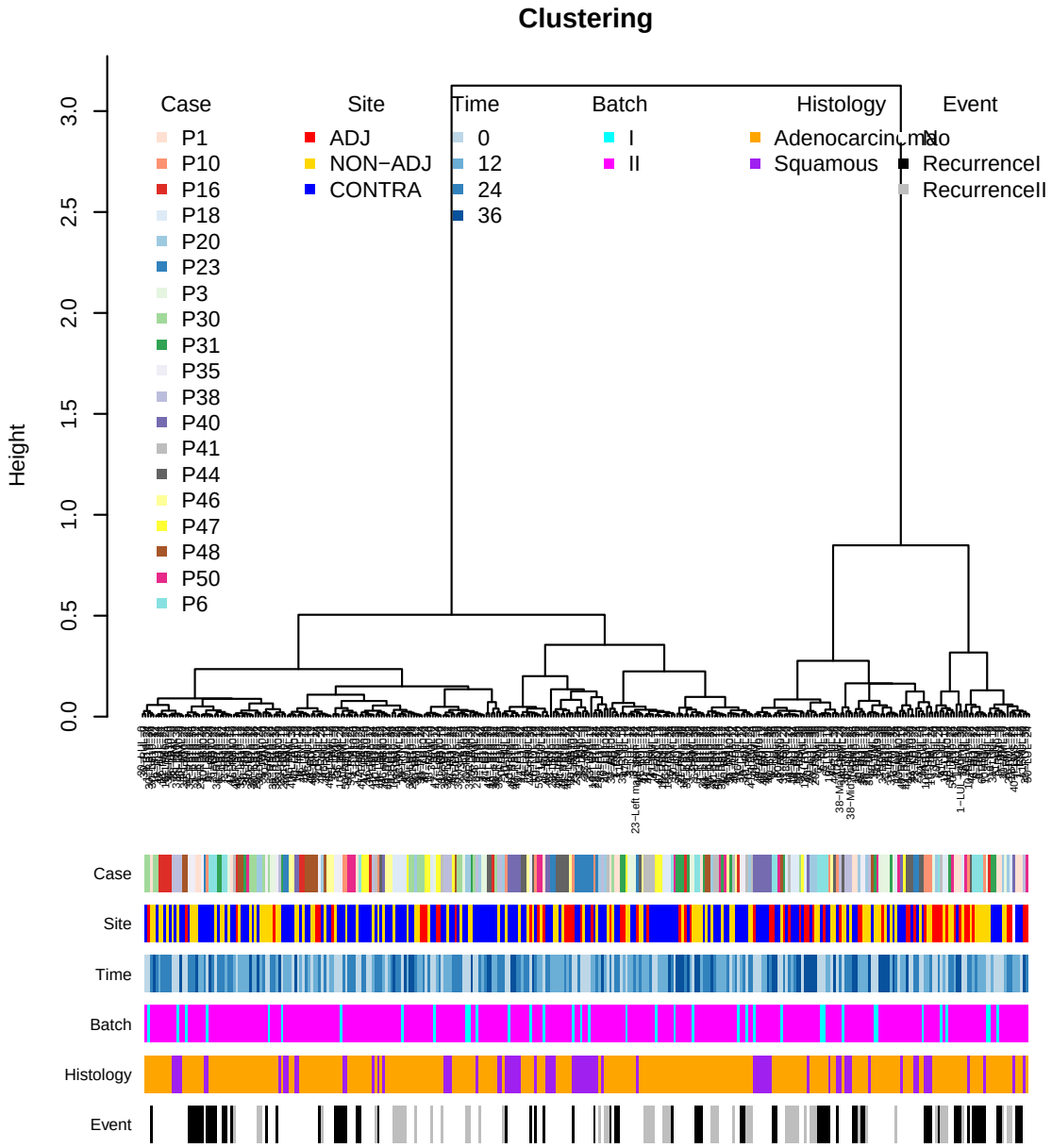

Figure 4: Hierarchical clustering of samples (using Pearson correlation and Ward's linkage) based on genes that are significantly different between **sites**.

We also want to cluster the genes that differ between sites. With both genes and samples clustered, we can construct a heatmap (**Figure 5**). The patterns in the heatmap suggest that there are at least two different gene expression patterns, which are indicated by the colorbar along the left side.

```
> ggc <- hclust(distanceMatrix(t(site.specific), "pearson"), "ward")
> scut <- cutree(ggc, k=2)
> scut.colors <- brewer.pal(3, "Dark2")[scut]
> sclass <- as.numeric(ssel)
> sclass[ssel] <- scut
> table(sclass)

sclass
    0     1     2
33116    23   113

> ssite <- t(scale(t(site.specific)))
> ssite[ssite > 5] <- 5
> ssite[ssite < -5] <- -5
```

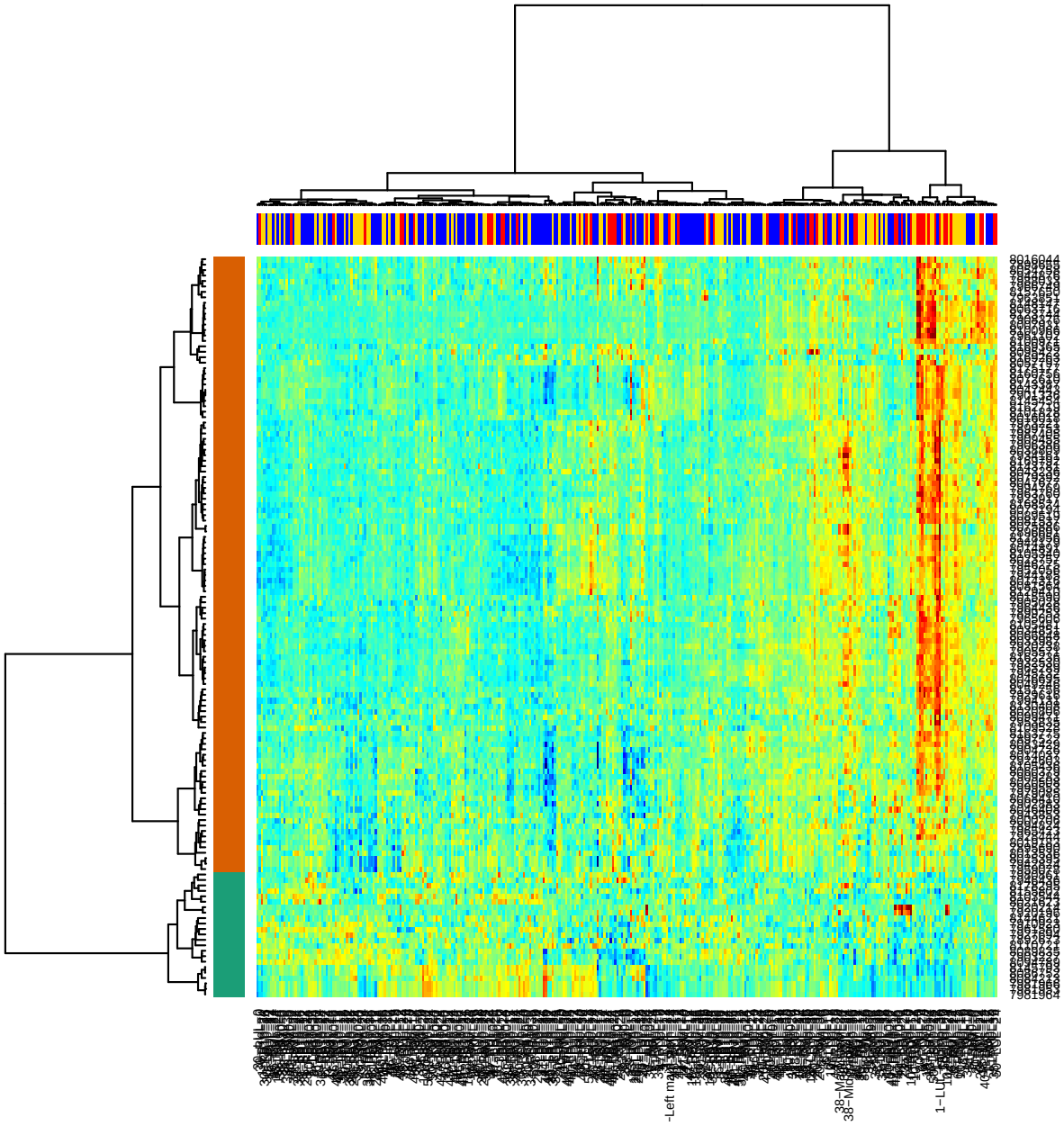

Figure 5: Two-dimensional clustering heatmap image of the genes selected because they differ by **site**. Top colorbar indicates site as in the previous plot. Left colorbar uses this clustering to define 2 types of gene expression patterns.

4.2 Genes That Change Over Time

In this section, we study genes that change (linearly) over time, regardless of the site. We start by selecting such genes with FDR equal to 5%.

```
> tsel <- selectSignificant(btime, alpha=0.05)
> time.lapse <- adjData[tsel,]
> gc <- hclust(distanceMatrix(t(time.lapse), "pearson"), "ward")
> tcut <- cutree(gc, k=8)
> tcut.colors <- brewer.pal(8, "Dark2")[tcut]
> tclass <- as.numeric(tsel)
> tclass[tsel] <- tcut
> table(tclass)

tclass
    0     1     2     3     4     5     6     7     8
30871   306   268   323   239   519   274   137   315

> results <- data.frame(time.lapse,
+                       Time.point=pvals[tsel, "Time.point"],
+                       TimeClass=tclass[tsel],
+                       annot[tsel,])
> results <- results[order(results$Time.point, decreasing=F),]
> write.csv(results, file=file.path(datadir, "time.lapse.csv"))
>
```

Next, we cluster the samples using these genes (**Figure 6**). The main branches are clearly unbalanced with respect to time; the starting time is much more likely to occur in the right-hand branch and the final time point is much more likely to appear in the left hand branch. In fact, over time, the samples seem to be moving from the right to the left branch.

```
> sc <- hclust(distanceMatrix(time.lapse, "pearson"), "ward")

> table(cutree(sc, k=2))

  1   2
240  91

> tab.time <- table(cutree(sc, k=2), si$Time.point)
> tab.time

     0 12 24 36
  1 44 69 78 49
  2 49 23 18  1
```

```
> round(tab.time[1,]/apply(tab.time, 2, sum)*100, 1)

   0   12   24   36
47.3 75.0 81.2 98.0

> fisher.test(tab.time)

        Fisher's Exact Test for Count Data

data:  tab.time
p-value = 1.67e-11
alternative hypothesis: two.sided
```

This effect becomes even more pronounced if we cut the tree slightly lower.

```
> table(cutree(sc, k=3))

  1   2   3
240  85   6

> tab.time2 <- table(cutree(sc, k=3), si$Time.point)
> tab.time2

     0 12 24 36
  1 44 69 78 49
  2 47 21 17  0
  3  2  2  1  1

> chisq.test(tab.time2)

        Pearson's Chi-squared test

data:  tab.time2
X-squared = 52.0489, df = 6, p-value = 1.823e-09
```

We also want to cluster the genes that differ between time. With both genes and samples clustered, we can construct a heatmap (**Figure 7**). The patterns in the heatmap suggest that there are at least eight different gene expression patterns, which are indicated by the colorbar along the left side.

```
> gc <- hclust(distanceMatrix(t(time.lapse), "pearson"), "ward")
> tcut <- cutree(gc, k=8)
> tcut.colors <- brewer.pal(8, "Dark2")[tcut]
> tclass <- as.numeric(tsel)
> tclass[tsel] <- tcut
> table(tclass)
```

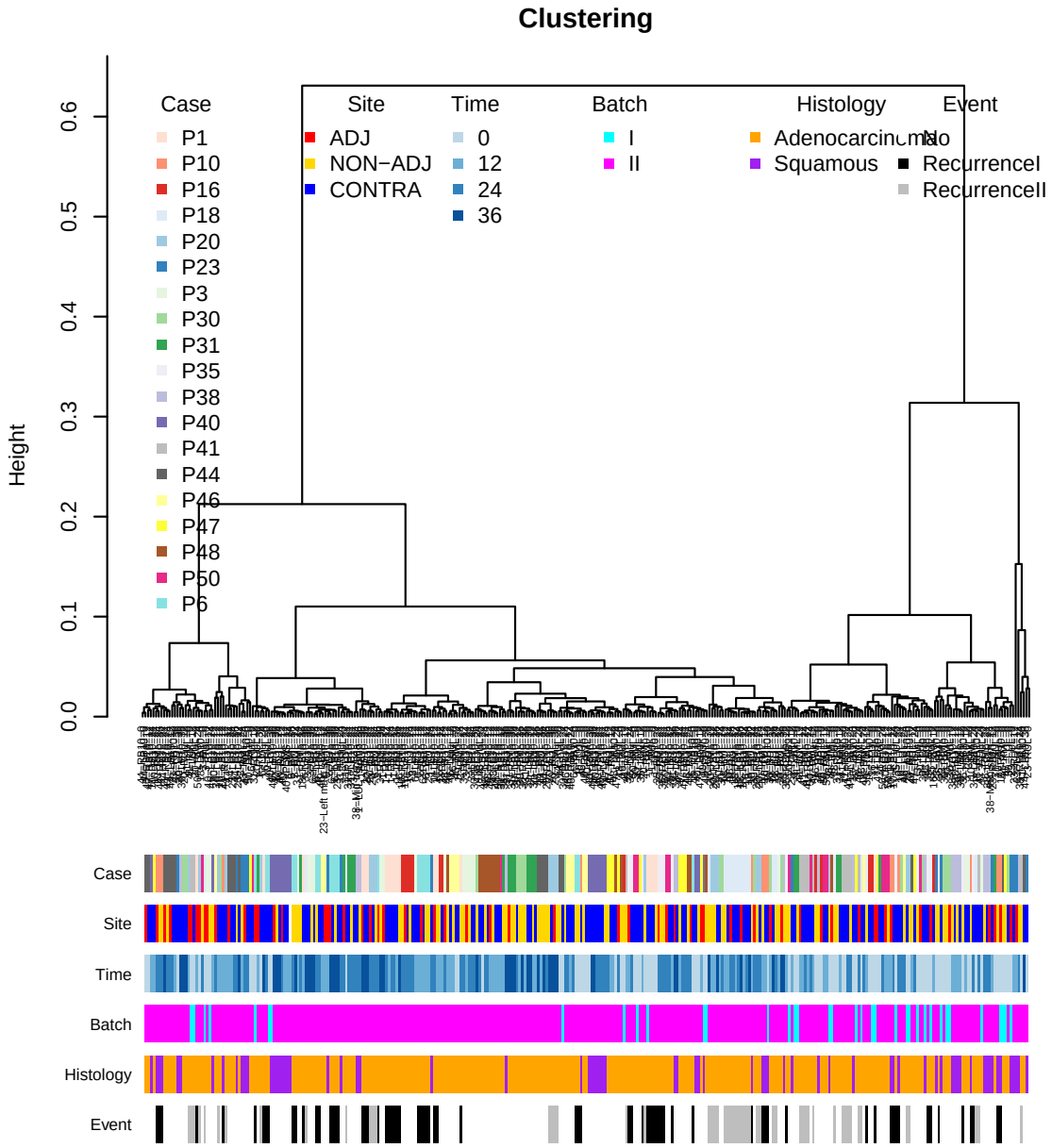

Figure 6: Hierarchical clustering of samples (using Pearson correlation and Ward's linkage) based on genes that are significantly different between **time points**.

```
tclass
    0     1     2     3     4     5     6     7     8
30871   306   268   323   239   519   274   137   315
```

```
> stime <- t(scale(t(time.lapse)))
> stime[stime > 5] <- 5
> stime[stime < -5] <- -5
```

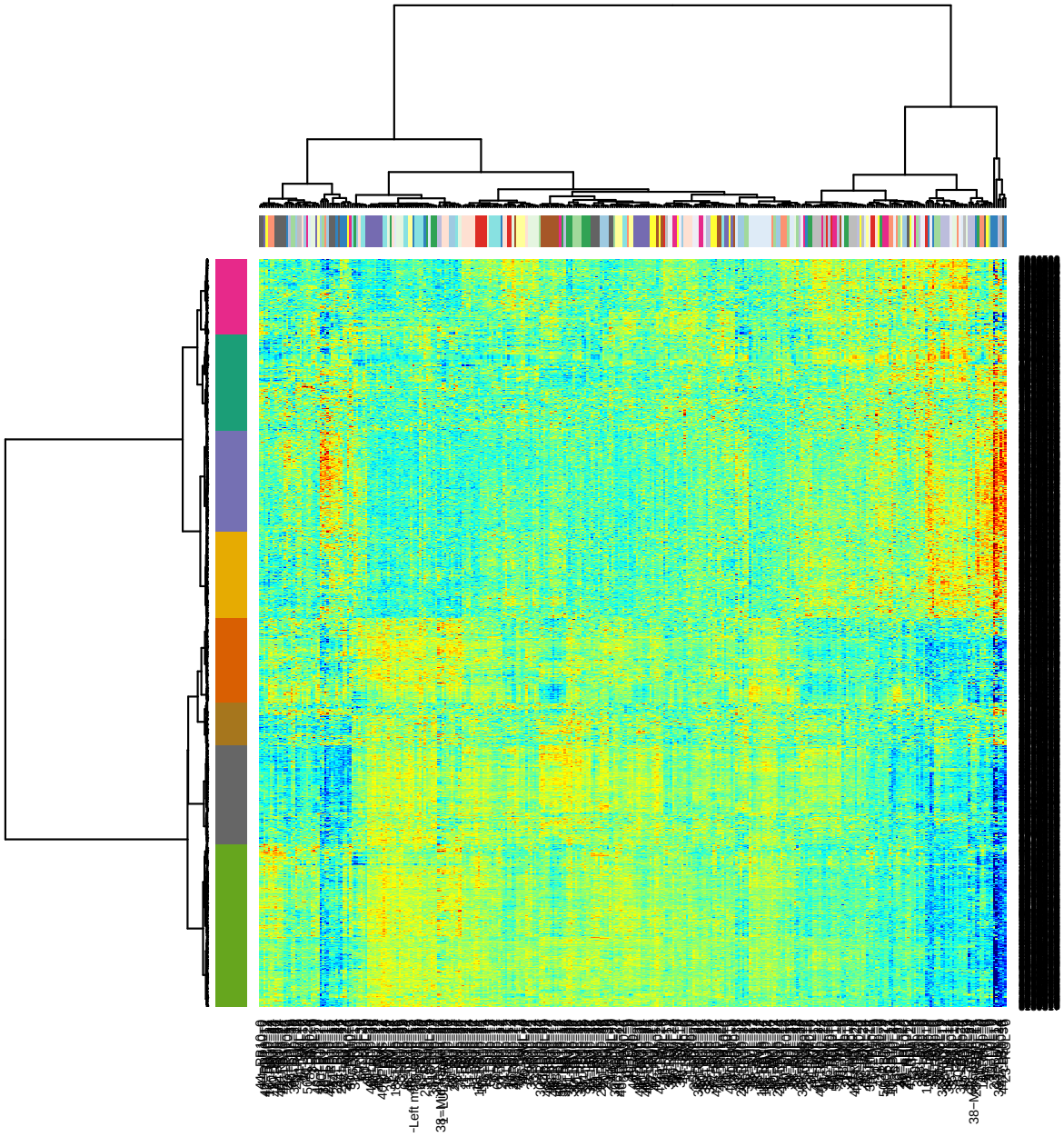

Figure 7: Two-dimensional clustering heatmap image of the genes selected because they differ by **time point**. Top colorbar indicates time, as in the previous plot. Left colorbar uses this clustering to define 8 types of gene expression patterns.

5 Overlapping Probabilities of Top Ranking Gene Lists using Hypergeometric tests

When the same set of genes appear in two top ranking gene lists in two different studies, it is often of interest to estimate the probability for this being a chance event. This overlapping probability is well known to follow the hypergeometric distribution.

Here we want to investigate whether the overlapping genes of genes selected from site and time is a chance event.

```
> load("01rev-krc.R")
> n  <- nrow(adjData)
> n1 <- nrow(site.specific)  # FDR 0.01
> n2 <- nrow(time.lapse)  # FDR 0.05
> m  <- length(intersect(rownames(site.specific), rownames(time.lapse)))
> c(n, n1, n2 ,m)

[1] 33252  1165  1395    42

> tab <- matrix(c(m, n2-m, n1-m, n-n1-n2+m), nrow = 2, byrow = FALSE)
> colnames(tab) <- c("InList1", "Not-InList1")
> rownames(tab) <- c("InList2", "Not-InList2")
> tab

            InList1 Not-InList1
InList2          42        1123
Not-InList2    1353       30734

> ### Hypergeometric Tests ###
> phyper(m-1,n1,n-n1,n2, lower.tail = FALSE)

[1] 0.864865

> phyper(min(n1,n2),n1,n-n1,n2, lower.tail = T) - phyper(m-1,n1,n-n1,n2, lower.tail = T)

[1] 0.864865

> (fisher.test(tab, alternative='greater'))$p.value

[1] 0.864865

>
```

Another set: overlapping genes between 238 most differentially expressed features selected at FDR 0.05 in difference of ADJ and CONTRA and 263 selected in cluster 1 of the site effect.

```
> n  <- nrow(adjData)
> n1 <- length(rownames(adjData)[which(sclass==1)])
> n2 <- length(rownames(read.csv(file="PairedTT-ADJvsCONTRA-BatchCorrect.csv",row.names=1)[1:23
> m  <- length(intersect(rownames(adjData)[which(sclass==1)],
+                        rownames(read.csv(file="PairedTT-ADJvsCONTRA-BatchCorrect.csv",row.nam
> c(n, n1, n2 ,m)

[1] 33252   263   238    66

> tab <- matrix(c(m, n2-m, n1-m, n-n1-n2+m), nrow = 2, byrow = FALSE)
> colnames(tab) <- c("InList1", "Not-InList1")
> rownames(tab) <- c("InList2", "Not-InList2")
> tab

            InList1 Not-InList1
InList2          66         197
Not-InList2     172       32817

> ### Hypergeometric Tests ###
> phyper(m-1,n1,n-n1,n2, lower.tail = FALSE)

[1] 5.919104e-84

> phyper(min(n1,n2),n1,n-n1,n2, lower.tail = T) - phyper(m-1,n1,n-n1,n2, lower.tail = T)

[1] 0

> (fisher.test(tab, alternative='greater'))$p.value

[1] 5.919104e-84

>
```

Another set: overlapping genes between 113 selected in cluster 2 of the site effect after removing main carinas and 263 selected in cluster 1 of the site effect.

```
> n  <- nrow(adjData)
> n1 <- length(rownames(adjData)[which(sclass==1)])
> n2 <- length(rownames(adjData)[which(rmMC.sclass==2)])
> m  <- length(intersect(rownames(adjData)[which(sclass==1)],
+                        rownames(adjData)[which(rmMC.sclass==2)]))
> c(n, n1, n2 ,m)

[1] 33252   263   113    96
```

```
> tab <- matrix(c(m, n2-m, n1-m, n-n1-n2+m), nrow = 2, byrow = FALSE)
> colnames(tab) <- c("InList1", "Not-InList1")
> rownames(tab) <- c("InList2", "Not-InList2")
> tab

            InList1 Not-InList1
InList2          96         167
Not-InList2      17       32972

> ### Hypergeometric Tests ###
> phyper(m-1,n1,n-n1,n2, lower.tail = FALSE)

[1] 2.45631e-191

> phyper(min(n1,n2),n1,n-n1,n2, lower.tail = T) - phyper(m-1,n1,n-n1,n2, lower.tail = T)

[1] 0

> (fisher.test(tab, alternative='greater'))$p.value

[1] 2.45631e-191

>
```

6 Appendix

We save the critical results.

```
> results <- data.frame(pvals,
+                       SiteClass=sclass,
+                       TimeClass=tclass,
+                       InteractionClass=iclass,
+                       annot)
> if (!file.exists("Output")) dir.create("Output")
> write.csv(results, file="results.csv")
```

This analysis was run in the following directory:

```
> getwd()

[1] "/data/bioinfo2/Lung-HN/Wistuba-VANGUARD/Analysis"
```

Note that '//mdadqsfs02/bioinfo2 is the standard insititutional location for storing data and analyses; 'O: is the name given to that location on this machine.

This analysis was run in the following software environment:

```
> sessionInfo()

R version 2.14.0 (2011-10-31)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=C                 LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] splines   stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] ClassDiscovery_2.10.2  mclust_3.4.11          cluster_1.14.2
 [4] ClassComparison_2.10.1 PreProcess_2.10.1      oompaBase_2.12.0
 [7] Biobase_2.14.0         RColorBrewer_1.0-5     lattice_0.20-6
[10] nlme_3.1-103

loaded via a namespace (and not attached):
[1] grid_2.14.0       KernSmooth_2.23-7 tools_2.14.0
```